基于金字塔结构的群智能演化策略求解混合整数规划问题 *

唐荷花，彭斯俊†，王占占

(武汉理工大学 理学院, 武汉 430070)

摘 要: 混合整数非线性规划问题（mixed-integer nonlinear programming, MINLP）广泛应用于科学及工程系统设计，传统的群智能算法在求解混合整数规划问题时，未能很好地解决种群内部个体或者种群之间开采与探索、竞争与协作的矛盾。为了解决这两个矛盾及更高效率地寻优，提出一种基于金字塔结构的群智能演化策略（swarm intelligent evolution strategy based on pyramid structure）的 PES 算法来求解混合整数规划问题。PES 算法中明确的分工机制能够平衡全局与局部搜索的能力，晋升机制解决了种群间竞争与协作的矛盾。利用标准测试函数进行仿真，对比改进的粒子群算法（CLSPSO、CLSPSO2）及改进的差分进化算法（ridDE、ridDE2）的结果，发现 PES 算法在成功率与精度方面具有优势，也体现了 PES 算法的有效性。

关键词: 非线性混合整数规划；竞争；协作；智能算法

中图分类号: TP18 doi: 10.19734/j.issn.1001-3695.2018.11.0805

Swarm intelligent evolution strategy based on pyramid structure for solving mixed integer programming problems

Tang Hehua, Peng Sijun†, Wang Zhanzhan

(*School of Science*, *Wuhan University of Technology*, *Wuhan* 430070, *China*)

Abstract: Mixed integer nonlinear programming (MINLP) is widely used in scientific and engineering system designs. The traditional swarm intelligent algorithm fails to solve the contradiction between mining and exploration, competition and cooperation among individuals or populations in the population when solving mixed integer programming problems. In order to solve these two contradictions and optimize more efficiently, this paper proposed a PES algorithm due to swarm intelligent evolution strategy based on pyramid structure to solve the mixed integer programming problems. PES algorithm had a clear division of labor mechanism and promotion mechanism. A clear division of labor could balance global and local search capabilities. The promotion mechanism resolved the contradictions between competition and collaboration among populations. Standard test functions were used to simulate, and the experimental results showed that success rate and accuracy of the PES algorithm are better and verify effectiveness of the PES algorithm by comparing the results of the improved particle swarm optimization algorithm (CLSPSO, CLSPSO2) and the improved differential evolution algorithm (ridDE, ridDE2) .

Key words: nonlinear mixed integer programming; competition; collaboration; intelligent algorithm

0 引言

在投资组合优化、运输问题、稀布天线阵列的优化设计、人工神经网络等实际工程中，存在着许多同时包含整数变量和连续变量的非线性优化问题，这类问题称为混合整数非线性规划问题（MINLP）。由于该问题同时含有整数变量和连续变量，以及目标函数和约束条件的非线性，使得结果通常存在局部最优解[1]，所以 MINLP 问题是一类 NP 完全问题。

求解 MINLP 的传统方法有分支定界法、扩展割平面法、外逼近法[2,3]等，但随着变量维数的增加，计算量会急剧增加，从而使这些方法存在很大的局限性。目前群智能演化算法广泛用于解决约束优化问题，且被证明很有效。近年来有许多学者将群智能演化算法，如遗传算法 (GA)、蚁群算法 (ACO)、粒子群优化算法 (PSO)和差分进化算法 (DE) 等[4~6]用于求解 MINLP 问题。这些方法一般都取得了满意的效果。但这些群智能演化算法(ACO、PSO、GA、DE 等)研究的核心是解决种群内部粒子开采与探索的矛盾。将研究的重心放在了竞争上，忽略了平衡种群内部个体或者种群之间竞争与协作的矛盾。以 PSO 算法为例，个体更新依赖于历史最优个体与当前最优个体对该个体产生的吸引力，这种吸引力主要表现为竞争，有少部分个体间的协作，并且这种协作是有偏好的协作，对协作的目的性不明确，使得个体间的信息交流和传递存在缺陷。

2014 年，Tan 等人[7]提出混沌粒子群优化与混沌搜索相结合来求解整数和混合整数规划问题；2011 年，张莉等人[8]提出一种嵌入正交杂交的差分进化算法，基于两水平正交表和因素分析的正交实验设计法所构成的正交杂交算子和差分进化的杂交算子相结合来增强差分进化的搜索性能；2017 年，赵乃刚等人[9]提出一种改进的蝙蝠算法，给出了一种自适应变化扰动步长，使算法能够更好地找出局部精确解，同时采用自然选择策略更新最差的部分解，提高种群的多样性；2017 年，赵佳鑫等人[10]提出一种基于随机游走的粒子群优化算法，

收稿日期: 2018-11-01；**修回日期:** 2018-12-11 **基金项目:** 国家自然科学基金资助项目（61573012）

作者简介: 唐荷花（1993-），女，河南许昌人，硕士研究生，主要研究方向为智能计算、统计分析；彭斯俊（1965-），男（通信作者），湖北仙桃人，教授，硕导，硕士，主要研究方向为演化计算、统计分析（sjpeng@whut.edu.cn）；王占占（1993-），男，安徽淮北人，硕士研究生，主要研究方向为智能计算.

通过构造出自适应的惯性权重，平衡了算法的全局和局部搜索能力。在这些改进算法中，文献[7~9]侧重于改进粒子的探索能力，文献[10]侧重于改进粒子的开采能力。对于种群中的个体，开采能力的增强伴随着探索能力的减弱。以上改进的算法对开采与探索之间的矛盾进行了一定程度的研究，但这些方法依旧是一些独立行为，没有将两者有机地统一。

因此，当前群智能演化算法的信息传递交换方法存在目的性不明确的弊端，未充分研究子种群之间的差异性，需要探寻一种合适的种群演化机制，使得种群个体之间在竞争与协作、开采与探索方面的需求达到平衡。2018 年，谈庆[11]提出的一种基于金字塔结构的群智能演化策略 PES（swarm intelligence evolution strategy based on pyramid structure），并基于这个策略设计了 PES 算法。PES 算法已成功应用于求解无约束的连续函数优化问题[11]，与其他的五种群智能算法（ACO、DE、ES、GA、PSO）相比，提高了算法的收敛速度且能有效避免早熟收敛。PES 算法简单，收敛速度快，有明确的分层机制和晋升机制，能够兼顾好种群内部与种群之间竞争与协作，平衡开采与探索的矛盾。鉴于 PES 算法的优点，本文利用 PES 算法来求解非线性混合整数规划这一实际问题，从而将 PES 算法的应用扩展到非线性混合整数规划问题的求解。

1　非线性混合整数规划问题

本文研究的非线性混合整数规划问题的一般模型如下：

$$\min_{x,y} z = f(X,Y)$$
$$s.t\begin{cases} h_i(X,Y)=0,\ i=1,2,...,r_h \\ g_j(X,Y)\le 0,\ j=1,2,...,r_g \\ x_i \in R^m \\ y_j \in Z^n \\ l_i \le x_i \le u_i \\ l_j \le x_j \le u_j \end{cases} \tag{1}$$

其中：$X=(x_1,x_2,...,x_m)$ 是 m 维的连续向量；$Y=(y_1,y_2,...,y_n)$ 是 n 维的整数向量；l_i 和 u_i 分别为变量 x_i 取值的上下界；l_j 和 u_j 分别为变量 x_j 取值的上下界；$f(X,Y)$ 为目标函数；$h_i(X,Y)$ 为等式约束条件，$g_j(X,Y)$ 为不等式约束条件；$f(X,Y)$，$h_i(X,Y)$，$g_j(X,Y)$ 至少有一个为非线性函数。

2　PES 算法求解非线性混合整数规划问题

2.1　PES 算法原理

PES 算法[11]是一种基于金字塔结构的群智能演化算法。该算法按照适应度函数值对种群进行分类得到子种群,根据分类的具体情况确定子种群开采、探索、传递三种不同的任务分工，进而构建出金字塔模型对函数优化问题进行建模求解的算法。

PES 算法有明确的分工机制。适应度最好的个体侧重于开采工作，适应度中等的个体负责开采、探索与传递工作，适应度较差的个体侧重于探索工作。明确的分工机制确保了在整个演化过程中，PES 算法能同时兼顾到开采与探索,起到平衡开采与探索矛盾的作用。

PES 算法提出了明确的晋升机制。底层经过随机搜索选出优秀个体传递给中间层,中间层对传上来的个体和自身的个体进行孵化操作，并将优秀的个体传递给顶层。这种层次递进的演化晋升操作不仅考虑了子种群内部个体存在的竞争关系，而且考虑子种群之间优秀个体存在的协作关系。明确的晋升机制确保了在整个演化过程中，PES 算法能同时兼顾竞争与协作,起到平衡竞争与协作矛盾的目的。

2.2　PES 算法步骤

a)参数设置。

设种群规模为 N，分层比例 D，传递比例 T，各层的搜索半径 R，加速步长 λ，控制收敛因子 α，初始可行解 x_0，最大迭代次数 $\max g$，并设置初始种群中可行解的比例。

b)初始化种群，并计算其适应度函数值。

采用实数和整数混合编码来表示个体，向量 $(x_t,y_t)=(x_{t1},x_{t2},...,x_{tm},y_{t1},y_{t2},...,y_{tn})$，表示种群中第 t 个个体，代表问题式(1)的一个解。

按式(2)(3)随机产生包含 N 个个体的初始种群，令 $g=1,t=1,2,...,N$。

$$x_{ti}=l_i+rand\times(u_i-l_i),(i=1,2,...,m) \tag{2}$$
$$y_{tj}=l_j+\text{randperm}(0,u_j-l_j),(j=1,2,...,n) \tag{3}$$

其中：$rand$ 为区间 [0,1] 均匀分布的随机数；$\text{randperm}(0,u_j-l_j)$ 表示 $[0,u_j-l_j]$ 内取得的随机整数。

c)根据适应度值进行分层操作。

随机产生的种群中的个体是杂乱无序的，分层操作使得个体间的分工更加明确，继而加快寻优的速度。

将初始种群中的个体 (x_t,y_t) 按照适应度值进行升序排列后，按照设置的分层比例 D 对个体进行分层：

$$[F,I]=sort(F_0) \tag{4}$$
$$P=P_0(I) \tag{5}$$

其中：F 是 F_0 经过升序操作后的适应度函数集；I 是升序时原初始种群适应度 F_0 对应的索引；P 是原初始种群 P_0 按照适应度 F_0 升序排列之后得到的种群。

下面对排序后的种群进行划分：

$$P=\{P_1,P_2,P_3,P_4\} \tag{6}$$

其中：$P_k(k=1,2,3,4)$ 分别表示开采层、两个传递层、探索层的子种群。

根据式(7)确定每层的个体数：

$$p_k=N\times D_k,(k=1,2,3,4) \tag{7}$$

其中：p_k 为第 k 层的个体数；D_k 为第 k 层的分层比率。

d)层内探索规则。

第一层(顶层)称为开采层。主要负责深入挖掘已存在的优秀个体所在区域；第二层和第三层称为传递层，它们的职责有三个：(a)兼并开采与探索的工作，由下至上开采能力逐步增强，探索能力逐步减弱；(b)通过传递操作建立开采层与探索层之间的桥梁；(c)对于传递上去的新的潜在优秀个体,需要进行保护培养，充分挖掘其潜力，不直接与该层中的其他个体竞争。第四层(最底层)称为探索层，探索层的搜索半径要大于开采层的搜索半径，因为探索层主要负责探索发现未知的潜在优秀个体及其所在区域，而开采层则着重于从已找到的优秀个体中找到更优秀的个体。

每层的随机探索规则为

$$\begin{cases} d=\min\{|x_{ti}-l_i|,|x_{ti}-u_i|,r_k\},(k=1) \\ r_1'=\delta\times d \\ x_{ti}'=x_{ti}\pm r_1',(k=1,i=1,2,...,m) \\ y_{tj}'=y_{tj}\pm R_k,(k=1,j=1,2,...,n) \\ x_{ti}'=x_{ti}+(2\times\delta-1)r_k,(k=2,3,4) \\ y_{tj}'=y_{tj}+randperm(2R_k+1,1)-R_k-1,(k=2,3,4) \end{cases} \tag{8}$$

其中：δ 为 [0,1) 均匀分布的随机数；$(x_t',y_t')=(x_{t1}',x_{t2}',...,x_{tm}',y_{t1}',y_{t2}',...,y_{tn}')$ 是第 k 层产生的新个体，(x_t,y_t) 为第 k 层已存在的个体；$randperm$ 表示随机产生在上下界范围内的整数。在开采层中，个体已处于适应度值较好的区域，个体探索的随机性减弱，因此减小搜索半径，处理后的半径为 r_1'。r_k 是第 k 层的连续变量搜索半径，R_k 是第 k 层的整数

变量搜索半径。整数变量的搜索半径为单位步长，式(8)其中对第 t 个个体整数变量的搜索策略为前进一步、后退一步或者原地不动。

连续变量的初始半径越往顶层越小，侧重于增强个体的开采能力，半径随着迭代次数增加而减小。

$$r_k = r_0 \times \alpha^g \ (k=1,2,3,4) \tag{9}$$

其中：r_0 为连续变量的初始半径；α 为控制收敛因子；g 为迭代次数。

由于新个体的向量 $(x_t^{'}, y_t^{'})$ 各分量 $x_{ti}^{'}(i=1,2,...,m)$，$y_{tj}^{'}(j=1,2,...,n)$ 可能落在搜索区间 $[l_i,u_i]$ 和 $[l_j,u_j]$ 外，所以对各分量进行检查并作修正，表示为

$$x_{ti}^{'} = \begin{cases} (-x_{ti}^{'} + l_i)/2 + l_i, & x_{ti}^{'} < l_i \\ x_{ti}^{'}, & l_i \le x_{ti}^{'} \le u_i \\ -(x_{ti}^{'} - u_i)/2 + u_i, & x_{ti}^{'} > u_i \end{cases} \tag{10}$$

$$y_{tj}^{'} = \begin{cases} ceil((l_j - y_{tj}^{'})/2) + l_j, & y_{tj}^{'} < l_j \\ y_{tj}^{'}, & l_j \le y_{tj}^{'} \le u_j \\ -ceil((y_{tj}^{'} - u_j)/2) + u_j, & y_{tj}^{'} > u_j \end{cases} \tag{11}$$

e)层内加速。

产生新点后，为了更好地实现初步筛选的目的，对新产生的个体进行加速操作：

$$\begin{cases} (x_t^s, y_t^s) = (x_t^{'}, y_t^{'}) + \lambda((x_t^{'}, y_t^{'}) - (x_t, y_t)), if\ f(x_t^{'}, y_t^{'}) \le f(x_t, y_t) \\ (x_t^s, y_t^s) = (x_t^{'}, y_t^{'}) - \lambda((x_t^{'}, y_t^{'}) - (x_t, y_t)), if\ f(x_t^{'}, y_t^{'}) > f(x_t, y_t) \end{cases} \tag{12}$$

其中：λ 为加速步长；(x_t^s, y_t^s) 为加速后新产生的个体；$f(x_t, y_t)$ 为第 t 个个体的适应度函数值，超出边界的加速点的处理方法同式(10)(11)。

f)层间传递。

在输送层产生新的个体后，从传输层确定的子种群中根据传递比率 T 按照规模为 2 的联赛机制向上层进行个体传输。接收层将传递上来的个体和自身已有个体一起进行孵化培养，充分挖掘优秀个体的潜力。

g)确定新种群。

将每层产生的新个体、加速个体、传递个体及培养过的个体，根据适应度值的优劣进行排序，并形成整合种群。从该种群中选择 p_k 个个体作为第 k 层的子种群，每个个体按照保留精英策略的单精度轮盘赌机制进行选择。先保留一个适应度值最好的个体，其他个个体根据式(13)[11]计算其被选择的概率。

$$P(x_t, y_t) = \begin{cases} \dfrac{2}{p_k(p_k - 1)} & t = p_k - 1 \\ P(x_{t+1}, y_{t+1}) + P(x_{p_k-1}, y_{p_k-1}) & t = 1,..., p_k - 2 \end{cases} \tag{13}$$

其中：$P(x_t, y_t)$ 表示排序后的第 t 个个体被选择的概率。设 (x_t, y_t) 是第 g 代种群依据式(6)得到的第 k 层的任意个体,按照层内探索、层内加速、层间传递、选择子种群四个步骤进行更新，将更新后的四个子种群整合在一起作为新种群。由于新种群中可能出现相同的个体，为了更好地实现种群多样性以及更有效率的寻优，只保留一个，其他个体用式(2)(3)随机产生的个体进行替代。找出当前种群中的最佳个体 *bestX* 及其目标函数值 *bestobjv* 。

h)停止条件。

如果达到最大迭代次数 $\max g$，则输出最优解 *bestX* 及最优值 *bestobjv*；否则令 $g = g + 1$，转步骤 c)。

3　仿真实验与分析

实验工具：MATLAB(R 2016a)。实验环境:Windows 7 操作系统，Intel 处理器 2.60 GHZ，4 GB 内存。

为了评估 PES 算法求解非线性混合整数规划问题的性能和有效性，本文对七个带约束的非线性混合整数规划问题的常用标准测试函数进行仿真[7,8]（见 3.1 节），该类函数是由实际的工程问题抽象而成，并且选取的函数具有很强的检验性。第一种测试函数连续性的变量多于离散性的变量，而第二种测试函数则相反，这样可以综合检验算法的性能。

将测试结果与改进的粒子群算法[7]和改进的差分进化算法[8,12]在最优值、最差值、平均值、方差和成功率多个方面进行分析比较。其中，改进的粒子群算法是从六个粒子群优化算法与混沌搜索相结合的其他六种混合算法（CLSPSO、CLSPSO1、CLSPSO2、CLSPSO3、CLSPSO4、CLSPSO5)中，选择每个问题测试效果最好的两种改进粒子群算法的结果，从而体现出 PES 算法在稳定性、成功率、最优解的精度等方面比改进的粒子群算法及改进的差分进化算法更具有优势。

3.1　测试函数

1）测试问题 1

$$\min f(x_1, x_2, y) = -y + 2x_1 + x_2$$
$$s.t \begin{cases} x_1 - 2\exp(-x_2) = 0 \\ -x_1 + x_2 + y \le 0 \\ 0.5 \le x_1 \le 1.4 \\ y \in \{0,1\} \end{cases}$$

最优解和最优值为 $(x_1, x_2, y; f) = (1.375, 0.375, 1; 2.124)$。

2) 测试问题 2

$$\min f(x_1, x_2, y) = -0.7y + 5(x_1 - 0.5)^2 + 0.8$$
$$s.t \begin{cases} -\exp(x_1 - 0.2) - x_2 \le 0 \\ x_2 + 1.1y \le -1 \\ x_1 - y \le 0.2 \\ 0.2 \le x_1 \le 1 \\ -2.22554 \le x_2 \le -1 \\ y \in \{0,1\} \end{cases}$$

最优解和最优值为 $(x_1, x_2, y; f) = (0.94194, -2.1, 1; 1.07654)$。

3) 测试问题 3

$$\min f(x_1, x_2) = (x_1 - 10)^3 + (x_2 - 20)^3$$
$$s.t \begin{cases} (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \ge 0 \\ -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \ge 0 \\ 13 \le x_1 \le 100, \text{integer} \\ 0 \le x_2 \le 100 \end{cases}$$

最优解和最优值为 $(x_1, x_2; f) = (15, 3.65464; -4242.00473)$。

4) 测试问题 4

$$\max f(x_1, x_2, x_3, y_1, y_2) = -5.357854x_1^2 - 0.835689y_1x_3 - 37.29329y_1 + 40792.141$$
$$s.t \begin{cases} a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3 \le 92 \\ a_5 + a_6y_2x_3 + a_7y_1y_2 + a_8x_1^2 - 90 \le 20 \\ a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 - a_{12}x_1x_2 - 20 \le 5 \\ 27 \le x_1, x_2, x_3 \le 45 \\ y_1 \in \{78, 79, ..., 102\}, \text{integer} \\ y_2 \in \{33, 34, ..., 45\}, \text{integer} \end{cases}$$

最优解和最优值为 $(x_1, x_3, y_1; f) = (27, 27, 78; 32217.4)$，$(x_2, y_2)$ 可以是各种不同的可行组合。问题 4 中各参数的取值如表 1 所示。

表 1　问题 4 中各参数的取值

Table 1　Value of each parameter in question 4

a_1 =85.334407	a_5 =80.51249	a_9 =9.300961
a_2 =0.0056858	a_6 =0.0071317	a_{10} =0.0047026
a_3 =0.0006262	a_7 =0.0029955	a_{11} =0.0012547
a_4 =0.0022053	a_8 =0.0021813	a_{12} =0.0019085

5) 测试问题 5

$$\min f(x)=\sum_{i=1}^{9}\left[\exp\left(-\frac{(u_i-x_2)^{x_3}}{x_1}\right)-0.01i\right]^2$$
$$where\ u_i=25+(-50\log(0.01i))^{2/3}$$
$$s.t\begin{cases}0.1\le x_1\le 100\\0\le x_2\le 25.6\\0\le x_3\le 5,\\x_1,x_2\ \text{integers}\end{cases}$$

最优解和最优值为 $(x_1,x_2,x_3;f)=(50,25,1.5;0)$ 。

6) 测试问题 6

$$\min f(x,y)=(y_1-1)^2+(y_2-1)^2+(y_3-1)^2-\ln(y_4+1)+(x_1-1)^2+(x_2-2)^2+(x_3-3)^2$$
$$s.t\begin{cases}y_1+y_2+y_3+x_1+x_2+x_3\le 5\\y_3^2+x_1^2+x_2^2+x_3^2\le 5.5\\y_1+x_1\le 1.2\\y_2+x_2\le 1.8\\y_3+x_3\le 2.5\\y_4+x_1\le 1.2\\y_2^2+x_2^2\le 1.64\\y_3^2+x_3^2\le 4.25\\y_2^2+x_3^2\le 4.64\\x_1,x_2,x_3\ge 0,y_1,y_2,y_3,y_4\in\{0,1\}\end{cases}$$

最优解和最优值为

$$(x_1,x_2,x_3,y_1,y_2,y_3)=(0.2,1.280624,1.954483,1,0,0,1)$$
$$f=3.557463$$

7）测试问题 7

$$\min f(x_1,x_2,x_3y)=x_1x_2x_3(y_1^2-y_1y_2+y_2^2)+x_1y_2^{0.5}+x_2$$
$$s.t\begin{cases}3x_1+x_2y_2=5\\5x_2+y_1\ge 6\\x_3+x_1\ge 1\\0\le y_1,y_2\le 2\\x_1,x_2,x_3\in\{0,1\}\end{cases}$$

最优解和最优值为

$$(x_1,x_2,x_3,y_1,y_2;f)=(1,1,0,1,2;1-\sqrt{2})$$

3.2　参数设置

实验中各个问题求解时，PES 算法的参数设置见表 2。其中问题 6 的最大迭代次数 $\max g$ 设为 70 000 代，运行次数为 10 次，其他问题的最大迭代次数均为 1 000 代，对每个问题独立运行 50 次。

表 2　求解问题的参数

Table 2　Parameters for solving proble

问题	参数					
	N	D	T	R	λ	α
P01	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.99
P02	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.99
P03	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.9999
P04	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.99
P05	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.99
P06	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.9999
P07	40	[0.1,0.2,0.3,0.4]	[0.2,0.4,0.6]	[0.01,0.05,0.1,0.2,1,1,1,1]	[0.1,1]	0.99

表中各参数的含义与 2.2 中各符号的定义相同。

3.3　实验结果及分析

记录运行 50 次后问题的最优解、对应的目标函数值、成功率(算法的成功率是指 50 次运行结果中目标函数值与已知目标函数值的差小于 0.1%的次数占比)，其中，问题 6 的运行次数设置为 10 次，统计结果见表 3。

表 3　本文算法的统计结果

Table 3　Statistical results of PES algorithm

问题	成功率/%	最优值	最优解
P01(min)	100	2.1245	(1.3748, 0.3748,1)
P02(min)	100	1.0765	(0.94194,-2.1,1)
P03(min)	100	-4242.0047	(15,3.65464)
P04(max)	100	32217.4278	(27,36.9427,27,78,39)
P05(min)	100	2.1622e-32	(50,25,1.5)
P06(min)	90	3.557463	(0.2, 1.28061258, 1.95448954, 1, 0, 0, 1)
P07(min)	100	-0.43444	(1,1,0,*,1.6667)

注：*在[1,2]之间任取

图 1～7 是问题 1～问题 7 最佳迭代次数运行结果的收敛曲线图。从所有问题的收敛曲线可以看到，大部分问题均能在 200 代找到最优解，并且前几代的收敛速度很快，这与种群之间的明确分工有密切关系。各算法运行结果如表 4 所示。

从表 4 可以看出，对于问题 P01～P05 和 P07，PES 算法均能在 1 000 代的进化过程中求出最优解；其中最小值问题 P07 和最大值问题 P04，本文求得的结果好于已知最优解，P07 目前已知最优解是[1,1,0,*,2]，其中*代表变量 y_i 的结果不唯一，在[1,2]取值。DE 算法求得的最优值和已知的最优值均为 $\sqrt{2}-1$，即-0.414214。而本文算法求得的最优解为[1,1,0,*,1.6667]，最小值为-0.43444，所求解优于目前所知最优解，表明了 PES 算法跳出局部最优的能力很强；从解的精度、最差值、平均值这三个指标来讲，对于问题 P01~P05，PES 算法求解的精度在一定程度上比 CLSPSO 算法和 CLSPSO2 算法所更高；从标准差的角度来看，标准差能够体现了算法的稳定性，方差越小，算法越稳定。PES 算法对问题 P01、P02、P04、P05、P06、P07 以最小的方差展现出算法的性能优势。在成功率这个指标来看，除问题 P06 的成功率为 90%外，稍逊于 ridDE 算法的成功率，其他测试问题均能达到 100%的成功率。

对于问题 P06，已知的最优值为 3.557463，最优解为(0.2,1.280624,1.954483,1,0,0,1)。对于此最优解，九个约束条件的值 $g_j(j=1,2,...,9)$ 分别为[-5.64893, 0.00000126665, 0, -0.51936, -0.545517, 0, -0.000002170624, -0.429996202711, -0.819996202711]，显然，该最优解违反了第二个约束条件，因此，此解是近似可行解，而非真正的最优解。本文算法通过增加迭代次数到七万代，得到最优值为 3.55746365455661，各变量的值为 (0.199999697952902, 1.28061258695081, 1.954489549651701,1,0,0,1)，此时约束条件的值为[-0.5649, -0.2123e-07, -0.3020e-08,-0.5194, -0.5455, -0.3020e-08, -0.3140e-06, -0.4300,-0.8200]。本文算法找到了一个真正可行的最优解。对于其他问题，本文算法均能有效地求解出最优解。PES 算法与改进的粒子群算法及改进的差分进化算法相比，对大部分测试函数，都能够有效提高解的精度、稳定性和成功率，具有明显的竞争优势。

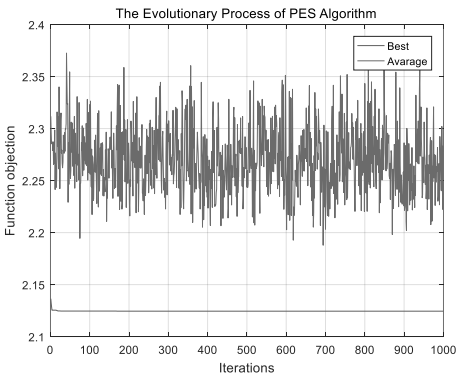

图 1　P01 的收敛曲线

Fig. 1　Convergence curve of P01

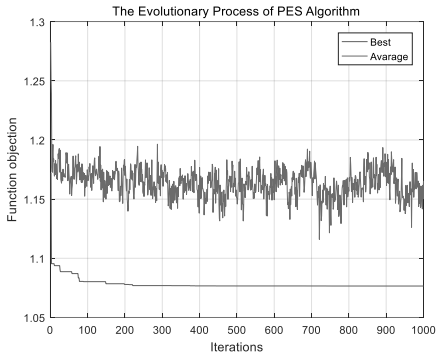

图 2　P02 的收敛曲线

Fig. 2　Convergence curve of P02

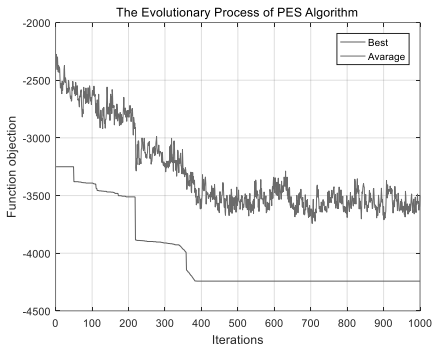

图 3　P03 的收敛曲线

Fig. 3　Convergence curve of P03

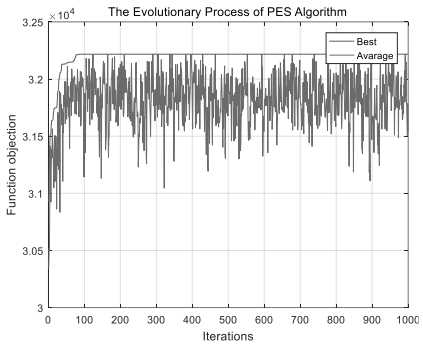

图 4　P04 的收敛曲线

Fig. 4　Convergence curve of P04

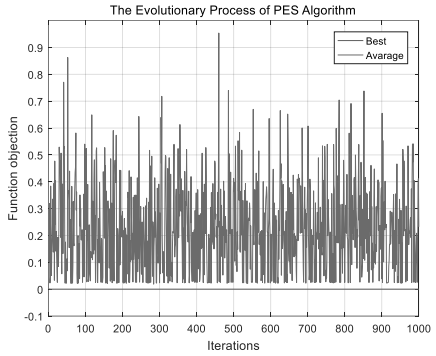

图 5　P05 的收敛曲线

Fig. 5　Convergence curve of P05

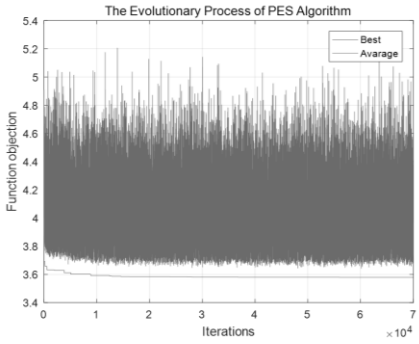

图 6　P06 的收敛曲线

Fig. 6　Convergence curve of P06

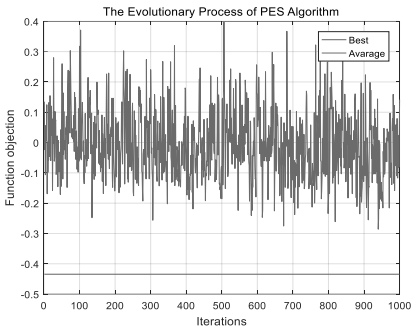

图 7　P07 的收敛曲线

Fig. 7　Convergence curve of P07

表 4　各算法运行结果

Table 4　Results of each algorithm operation

测试问题（已知最优解）	算法	最优值	最差值	平均值	标准差	成功率（%）
P01（2.124）	**PES**	**2.1245**	**2.1245**	**2.1245**	**2.5591e-09**	**100**
	CLSPSO	2.1245	2.1245	2.1245	1.71e-07	100
	CLSPSO2	2.1245	2.1245	2.1245	1.3756e−05	100
P02（1.07654）	**PES**	**1.0765**	**1.0765**	**1.0765**	**1.8335e-07**	**100**
	CLSPSO	1.0765	1.0894	1.0771	0.0020	86
	CLSPSO2	1.0766	1.0828	1.0790	0.0017	28
P03（-4242.00473）	**PES**	**-4242.0047**	**-4241.9754**	**-4241.9993**	**0.00547**	**100**
	CLSPSO	-4242	-4242	-4242	3.67e-12	100
	CLSPSO2	−4242	−4241.8	-4242	0.0557	100
P04（32217.42）	**PES**	**32217.4278**	**32217.4278**	**32217.4278**	**1.819e-11**	**100**
	CLSPSO	32217	32217	32217	2.20e-11	100
	CLSPSO2	32217	32158	32210	19.6487	88
P05（0）	**PES**	**2.1622e-32**	**2.3646e-06**	**3.3177e-07**	**4.0997e-07**	**100**
	CLSPSO	3.7871e-08	3.4151e-04	1.1893e-05	5.2502e-05	100
	CLSPSO2	4.6253e-08	3.5139e-04	1.8628e-05	5.4402e-05	100
P06（3.557463）	**PES**	**3.557463**	**3.560098**	**3.579559**	**0.0065611**	**90**
	ridDE	3.557465	---	3.557513	0.007497	100
P07（$1-\sqrt{2}$）	**PES**	**-0.43444**	**-0.43444**	**-0.43444**	**1.3801e-14**	**100**
	ridDE2	-0.414214	-0.414214	-0.414214	---	100

4　结束语

针对非线性混合整数规划问题，本文利用一种基于金字塔结构的群智能演化策略的 PES 算法进行求解，通过解决种群或个体之间的开采与探索、竞争与协作的两个矛盾，使得它比传统的一些群智能演化算法有着更好的跳出局部最优的寻优性能。数值结果表明，PES 算法不仅是有效的，而且是稳定的，该算法为求解 MINLP 问题提供了全新的思路。但它仍有许多方面需要进行改进，比如 PES 算法的参数较多，能否通过找到一组最优参数来提高算法效率。下一步可以延拓到 PES 算法是否能够应用于求解组合优化问题，如 TSP 问题、0-1 背包问题等混合整数规划问题。

参考文献：

[1] 黄玉兰，夏璞，夏岩，等. 物联网射频识别 (RFID) 核心技术详解 [M]. 北京：人民邮电出版社，2012: 22-25. (Huang Yulan, Xia Wei, Xia Yan, *et al.* Detailed explanation of the core technology of internet of things radio frequency identification (RFID) [M]. Beijing: People's Postsand Telecommunications Press, 2012: 22-25.)

[2] 马艳利. 混合整数非线性规划问题的分支定界算法研究 [D]. 宁夏：宁夏大学，2014. (Ma Yanli. Research on branch and bound algorithm for mixed integer nonlinear programming problems [D]. Ningxia: Ningxia University, 2014.)

[3] 仝哲，张炳江，李慧. 关于存在多组最优解的整数线性规划问题的割平面法的研究 [J]. 数学的实践与认识，2017，47 (5): 158-164. (Tong Zhe, Zhang Bingjiang, Li Hui. Study on the cutting plane method for integer linear programming problems with multiple optimal solutions [J]. Mathematics in Practice and Theory, 2017, 47 (5): 158-164.)

[4] 张章，汪亚明，郑俊褒，等. 混沌遗传算法用于求解混合整数规划问题 [J]. 工业控制计算机，2015 (4): 123-124. (Zhang Zhang, Wang Yaming, Zheng Junbao, *et al.* Chaos genetic algorithm for solving mixed integer programming problems [J]. Industrial Control Computer, 2015 (4): 123-124.)

[5] 谭跃，谭冠政，邓曙光. 基于遗传交叉和多混沌策略改进的粒子群优化算法 [J]. 计算机应用研究，2016，33 (12): 3643-3647. (Tan Yue, Tan Guanzheng, Deng Shuguang. Improved particle swarm optimization algorithm basedon genetic crossover and multichaos strategy [J]. Application Researchof Computers, 2016, 33 (12): 3643-3647.)

[6] Wu Jun, Gao Yuelin, Yan Lina. An improved differential evolution algorithm for mixed integer programming problems [C]// Proc of the 9th International Conference on Computational Intelligence & Security. Washington DC:IEEE Computer Society, 2013: 31-35.

[7] Tan Yue, Tan Guanzheng, Deng Shuguang. Hybrid particle swarm optim-ization with chaotic search for solving integer and mixed integer pro-gramming problems [J]. Journal of Central South University, 2014, 21 (7): 2731-2742.

[8] 张莉，李宏，冯大政. 求解混合整数规划的嵌入正交杂交的差分进化算法 [J]. 系统工程与电子技术，2011，33 (9): 2126-2132. (Zhang Li, Li Hong, Feng Dazheng. Differential evolutionary algorithm for embedded orthogonal hybridization based on mixed integer programming [J]. System Engineering and Electronics, 2011, 33 (9): 2126-2132.)

[9] 赵乃刚，李勇. 基于改进蝙蝠算法的混合整数规划问题 [J]. 微电子学与计算机，2017，34 (6): 94-98. (Zhao Naigang, Li Yong. Hybrid integer program-ming problem based on improved bat algorithm [J]. Microelectronics & Computer, 2017, 34 (6): 94-98.)

[10] 赵佳鑫，高岳林，陈群林. 一种求解非线性规划问题的粒子群算法 [J]. 宁夏大学学报：自然科学版，2017，38 (1): 15-18,22. (Zhao Jiaxin, Gao Yuelin, Chen Qunlin. A particle swarm optimization algorithm for solving no-nlinear programming problems [J]. Journal of Ningxia University: Nat-ural Science, 2017, 38 (1): 15-18,22.)

[11] 谈庆. 基于金字塔结构的群智能演化策略 [D]. 武汉：武汉理工大学，2018. (Tan Qing. A swarm intelligent evolution strategy based on pyramid structure [D]. Wuhan: Wuhan University of Technology, 2018.)

[12] Datta D, Figueira, José Rui. A real-integer-discrete-coded differential evolution [J]. Applied Soft Computing, 2013, 13 (9): 3884-3893.